IEICE TRANS. INF. & SYST., VOL.E93–D, NO.3 MARCH 2010

LETTER *Special Section on Trust, Security and Privacy for Pervasive Applications*

On the Security of RFID Group Scanning Protocols

Duc Nguyen DANG[†a)], ***Nonmember and*** **Kwangjo KIM**[†], ***Member***

SUMMARY A RFID group scanning protocol enables a RFID reader to produce a proof of co-existence of multiple RFID tags. This type of protocol is also referred to as yoking-proof, grouping-proof and co-existence proof. In this letter, we show that all of the previous group scanning protocols are vulnerable to relay attack.

key words: *yoking-proof, grouping-proof, co-existence-proof, relay attack*

1. Introduction

Juels introduced the concept of RFID group scanning by presenting a protocol called yoking-proof [1]. The protocol enables a reader to produce a proof of co-existence of two tags within its communication range. Unfortunately, yoking-proof is vulnerable to replay attack [2]. The improved versions were presented in [2]–[5] to resist against replay attack. In this letter, we present a universal relay attack on all RFID group scanning protocols in [1]–[5]. Our attack is practical in a sense that an attacker only needs to relay messages between a reader and a genuine. As a result of our attack, the reader produces a valid co-existence proof but containing a tag that is not supposed to be scanned.

2. Relay Attacks on Group Scanning Protocols

Throughout this letter, we will use notations summarized in Table 1.

We first describe our attacking model as follows: the attacker acts as a proxy between a reader and a genuine tag which is out of the communication range of the reader. The attacker then relays messages exchanged between the reader and the victim tag so that the resulting co-existence proof contains the victim tag. Our attack is sometimes referred to as *mafia fraud attack*. The attacking model is illustrated in Fig. 1.

We now briefly review RFID group scanning protocols in [3]–[5] and present our relay attacks on those protocols.

2.1 Relay Attack on Piramuthu's Protocol

Piramuthu's protocol [3] is an improved version of yoking-proof which addresses vulnerability of the timestamp-based yoking-proof proposed in [2]. The protocol proceeds as follows:

P1. $R \rightarrow T_1$: r chosen at random.
P2. $T_1 \rightarrow R$: T_1, r_1 chosen at random.
P3. $R \rightarrow T_2$: r, r_1.
P4. $T_2 \rightarrow R$: T_2, r_2 and m_2 where r_2 is randomly chosen and $m_2 = \mathrm{MAC}_{K2}[r, r_1]$.
P5. $R \rightarrow T_1$: m_2.
P6. $T_1 \rightarrow R$: $m_1 = \mathrm{MAC}_{K1}[r_1, m_2]$.
P7. $R \rightarrow V$: $P = (r, r_1, T_1, m_1, r_2, T_2, m_2)$.

Our attack works as follows: the attacker renders itself as the tag T_2, which is out of the communication range of the reader, in the above protocol. The attacker starts to intervene in step P3 as follows:

P3. $R \rightarrow$ Attacker: r, r_1.
P3′. Attacker $\rightarrow$ T2: r, r_1.
P4′. T2 $\rightarrow$ Attacker: T_2, r_2, m_2.
P4. Attacker $\rightarrow R$: T_2, r_2, m_2.

Table 1 Notations.

Notation	Description
c	Counter value stored in a tag
$f(.)$	A pseudorandom function
K_i	Secret of tag T_i
$\mathrm{MAC}_K[.]$	Message authentication code with secret K
P	A co-existence proof of multiple tags
R/V	Reader/Verifier (Backend Database)
$\mathrm{SK}_K[.]$	Symmetric encryption with secret K
T_1, T_2	RFID Tags
TS	Timestamp
x	Secret of Verifier

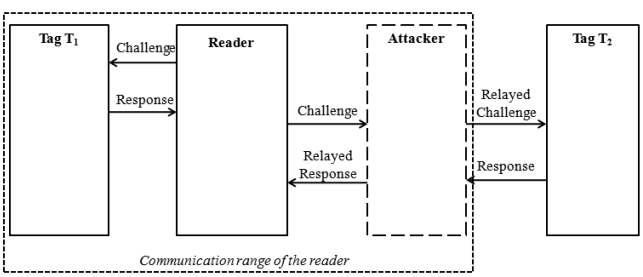

Fig. 1 Our attacking model.

Manuscript received June 30, 2009.
Manuscript revised October 6, 2009.

[†]The authors are with CAIS Lab, Department of Information and Communications Engineering, KAIST, Republic of Korea.
a) E-mail: nguyenduc@icu.ac.kr

DOI: 10.1587/transinf.E93.D.528

2.2 Relay Attack on Lin et al.'s Protocol

Lin et al. proposed two protocols using timestamp to address a race condition in Piramuthu's protocol where multiple readers are present [4]. The main idea of Lin et al.'s protocol is to encrypt the timestamp before sending to the reader so that attackers cannot collect MAC on different timestamp values from tags. Our attack on their first protocol is presented here but it is also applicable to the second one. The first protocol in [4] proceeds as follows:

L1. $\mathrm{V} \rightarrow \mathrm{R}$: $\mathrm{S} = \mathrm{SK}_x[r, \mathrm{TS}]$ where r is randomly chosen.
L2. $\mathrm{R} \rightarrow \mathrm{T}_1$: S.
L3. $\mathrm{T}_1 \rightarrow \mathrm{R}$: T_1, $m_1 = \mathrm{MAC}_{\mathrm{K1}}[\mathrm{S}]$.
L4. $\mathrm{R} \rightarrow \mathrm{T}_2$: S, m_1.
L5. $\mathrm{T}_2 \rightarrow \mathrm{R}$: $\mathrm{T}_2, m_2 = \mathrm{MAC}_{\mathrm{K2}}[\mathrm{S}, m_1]$.
L6. $\mathrm{R} \rightarrow \mathrm{V}$: $\mathrm{P} = (\mathrm{S}, \mathrm{T}_1, m_1, \mathrm{T}_2, m_2)$.

Our attack on Lin et al.'s protocol involves relaying messages in steps L4 and L5 as follows:

L4. $\mathrm{R} \rightarrow$ Attacker: S, m_1.
L4′. Attacker $\rightarrow \mathrm{T}_2$: S, m_1.
L5′. $\mathrm{T}_2 \rightarrow$ Attacker: T_2, m_2.
L5. Attacker $\rightarrow \mathrm{R}$: T_2, m_2.

Note that, Lin et al.'s protocols use timestamp to verify a proof. However, as we can assume that the processing time of the attacker is much faster than that of a tag, the lifespan of the protocols under attack can still be correctly verified.

2.3 Relay Attack on Burmester et al.'s Protocol

Burmester et al. also proposed two protocols for RFID group scanning, one with and one without tag anonymity [5]. The authors assume the two tags in a group share a common group id *gid* and a common secret key K_g. The protocol without tag anonymity is described below.

B1. $\mathrm{R} \rightarrow \mathrm{T}_1$, T_2: r chosen at random.
B2. T_1, $\mathrm{T}_2 \rightarrow \mathrm{R}$: *gid*.
B3. $\mathrm{R} \rightarrow \mathrm{T}_1$, T_2: T_1 and T_2 are linked.
B4. $\mathrm{T}_1 \rightarrow \mathrm{R}$: c, r_1 where $r_1 \| s_1 = f(r, c, \mathrm{K}_g)$.
B5. $\mathrm{R} \rightarrow \mathrm{T}_2$: r_1, c.
B6. $\mathrm{T}_2 \rightarrow \mathrm{R}$: t_2, s_2 if $r_1 = r_2$ where $r_2 \| s_2 = f(r, c, \mathrm{K}_g)$ and $t_2 = f(r_2, c, \mathrm{K}_2)$. If $r_1 \neq r_2$, T_2 terminates the protocol.
B7. $\mathrm{R} \rightarrow \mathrm{T}_1$: s_2.
B8. $\mathrm{T}_1 \rightarrow \mathrm{R}$: t_1 if $s_1 = s_2$ where $t_1 = f(r_1, c, \mathrm{K}_1)$. T_1 also update its counter value $c = c+1$. If $s_1 \neq s_2$, T_1 terminates the protocol.
B9. $\mathrm{R} \rightarrow \mathrm{V}$: $\mathrm{P} = (r, gid, c, r_1, t_1, r_2, t_2)$.

Our attack on the above protocol can be executed in the same fashion as before. The attacker intervenes in the steps B1, B2, B3, B5 and B6 to relay the corresponding messages between the reader and the tag T_2. Note that, the attacker needs to know two tags having the same group id beforehand. However, tags that have a common group id tend to be physically close to each other. In addition, it is trivial to collect group ids of tags. Therefore, our relay attack is still effective on Burmester et al.'s protocols.

3. Relay Attack versus Replay Attack

We compare relay attack and replay attack as follows:

- Relay attack is about faking a proof with forged tag location whereas replay attack can be about both forged location and time. It is because the relay attacker has to communicate with a victim tag and a reader whenever he wishes to create a forged proof. It is not the case for the replay attacker who reuses information obtained in previous sessions to create a forged.
- The relay attacker does not actively modify any message when communicating with tags and readers. On the other hand, the replay attacker may need to make adaptive queries to legitimate tags and readers in order to obtain information that is required to forge a proof.

4. Countermeasure and Conclusion

In order to prevent relay attack, we should prevent tag location from being forged. One can use a so called distance-bounding protocol to verify the tag location by measuring time taken by one querying session. We can implement a distance-bounding protocol for a grouping-proof protocol as follows:

- R: Start clock.
- $\mathrm{R} \leftrightarrow \mathrm{T}_i$: Query the tag T_i.
- R: Stop clock; Include T_i in the proof only if T_i's response is received within a pre-defined amount of time.

In conclusion, we have presented a universal relay attack on current grouping-proof protocols and one countermeasure. We think that it is important to address relay attack when defining a security notion for a secure grouping-proof protocol. Otherwise, security proof cannot be achieved.

Acknowledgement

This work was supported by the IT R&D program of MKE/TTA, 2009-P1-14-08I91, Mobile and next generation RFID technology standards development.

References

[1] A. Juels, "Yoking-proofs for RFID tags," Proc. First International Workshop on Pervasive Computing and Communication Security, pp.138–143, 2004.

[2] J. Saito and K. Sakurai, "Grouping-proofs for RFID tags," Proc. AINA International Conference, pp.621–624, 2005.
[3] S. Piramuthu, "On existence proofs for multiple RFID tags," Proc. ACS/IEEE International Conference on Pervasive Services, pp.317–320, 2006.
[4] C.-C. Lin, Y.-C. Lai, J.D. Tygar, C.-K. Yang, and C.-L. Chiang, "Coexistence proof using chain of timestamps for multiple RFID tags," Proc. APWeb/WAIM International Workshop, Springer-Verlag, LNCS 5189, pp.634–643, 2007.
[5] M. Burmester, B. de Medeiros, and R. Motta, "Provably secure grouping-proofs for RFID tags," Proc. CARDIS International Conference, Springer-Verlag, LNCS 5189, pp.176–190, 2008.